EL-GY 6483: REAL TIME EMBEDDED SYTEMS

NYU TANDON SCHOOL OF ENGINEERING
Spring 2025

EL-GY 6483: REAL TIME EMBEDDED SYTEMS

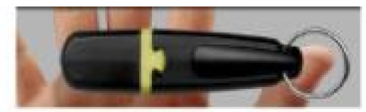
- Instructor:
  - Professor Matthew Campisi
    - Office: 370 Jay Street, 8th Floor
    - Email: mcampisi@nyu.edu
  - Teaching Assistants:
    - TBD

Logistics

- Class Meeting Times and Locations:
  - 370 Jay Street Rm. 202/ZOOM
- Textbook – No required Text, but many reference materials..
- Quizzes:
  - 2 Open Book
- Embedded Challenge Spring 2025
  - TBD (More info discussed in class)

- Grading Policy
  - Exams 40% (20% each)
  - Homework (Given weekly) 20%
  - Embedded Challenge 40%

EMBEDDED SYSTEMS



EMBEDDED SYSTEMS

What are Embedded Systems?

- Anything that uses a microprocessor but isn't a general-purpose computer
  - Smartphones?
  - Set-top boxes
  - Televisions
  - Video Games
  - Refrigerators
  - Cars
  - Planes
  - Elevators
  - Remote Controls
  - Alarm Systems

- The user "sees" a smart (***special-purpose***) system as opposed to the computer inside the systems
  - "how does it do that?"
  - "it has a computer inside of it"
  - "oh, BTW, it does not or cannot run Windows or MacOS"

- The end-user typically does not or cannot modify or upgrade the internals

What about this course?

- **Hardware**
  - I/O, peripherals: sensors/actuators, memory, busses, devices, control logic, interfacing hardware to software

- **Software**
  - Lots of C and assembly, device drivers, low-level real-time issues, scheduling
  - Concurrency; interrupts

- **Software/Hardware interactions**
  - Where is the best place to put functionality – hardware or software?
  - What are the costs:
    - Performance
    - Memory requirements (RAM and/or FLASH ROM)

- **Integration of hardware and software**
  - Programming, logic design, architecture
  - Algorithms, mathematics and *common sense*

Careers in Embedded?

- Automotive systems
  - Perhaps designing and developing “drive-by-wire” systems
  - Self-driving vehicles

- Telecommunications

- Medical Devices

- Consumer electronics
  - Cellular phones, MP3 devices, integrated cellular/tablet
  - Set-top box and HDTV
  - Home and Internet appliances/ IOT
    - Your refrigerator will be on the internet more than you are!

- Defense and weapons systems

- Process control
  - Gasoline processing, chemical refinement

- Automated manufacturing
  - Supervisory Control and Data Acquisition (SCADA)

- Space communications
  - Satellite communications

Goals of the Course

- **High-level Goals**
  - Understand the scientific principles and concepts behind embedded systems
  - Obtain hands-on experience in programming embedded systems

By the end of the course, you should be able to

- Understand the “big ideas” in embedded systems
- Obtain direct hands-on experience on both hardware and software elements commonly used in embedded systems design
- Understand the basics of embedded system application concepts such as signal processing and feedback control
- Understand and be able to discuss and communicate intelligently about:
  - Embedded processor architecture and programming
  - OS primitives for concurrency, timeouts, scheduling, communication and synchronization

The Big Ideas

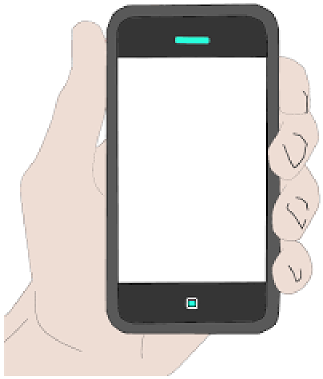
- HW/SW Boundary
- Non processor centric view of architecture
- Bowels of the operating software
  - Specifically, basic real-time operation with interrupts
  - Concurrency
- Real-world design
  - Performance vs. cost tradeoffs
- Analyzability
  - How do you “know” that your drive-by-wire system will function correctly?
- Application-level techniques
  - Signal processing, control theory
  - Semaphores, locks, atomic sections

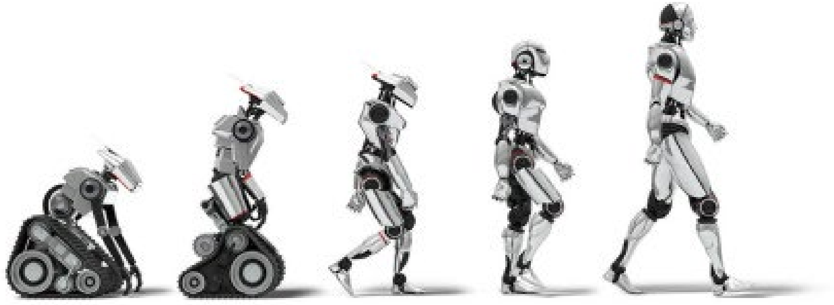
What is an Embedded system?

- An **embedded system** is a computer **system** with a dedicated function within a larger mechanical or electrical **system**, often with real-time computing constraints. It is **embedded** as part of a complete device often including hardware and mechanical parts. **Embedded systems** control many devices in common use today.

- Typically dedicated software (may be user customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general purpose display device

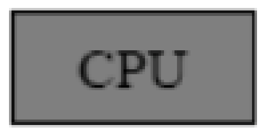
However, every system is unique – there are always exceptions





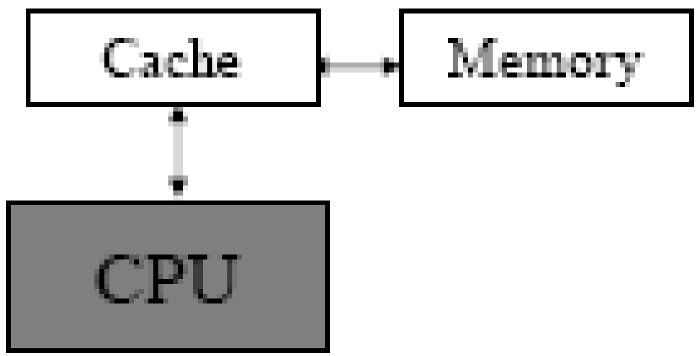
CPU: An All-Too-Common View of Computing

- Measured by:
  - Performance



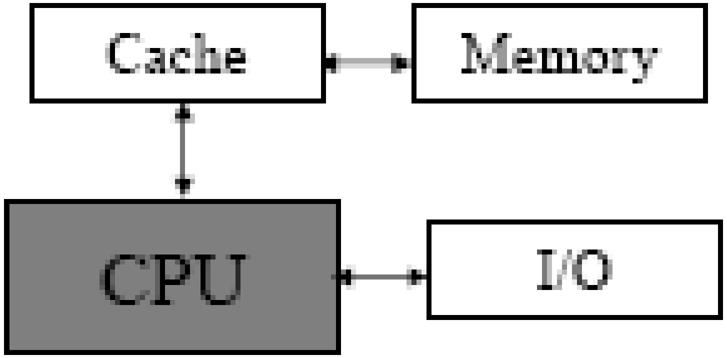
An Advanced Computer Engineer’s View

- Measured by: Performance
  - Compilers matter too…



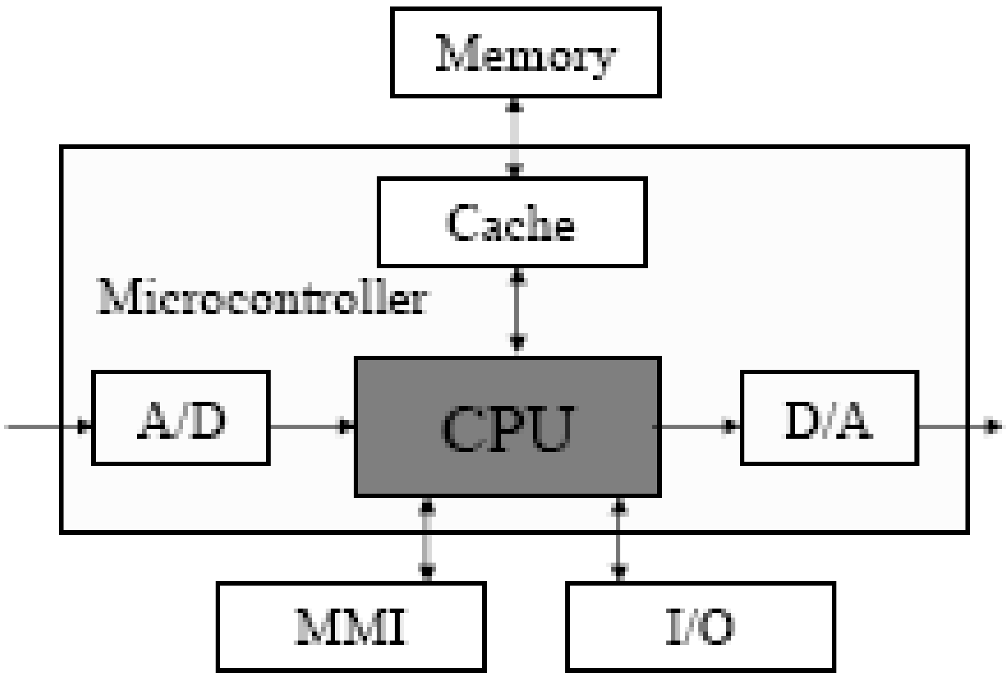
An Enlightened Computer Engineer's View

- Measured by: Performance, Cost
  - Compilers & OS matters



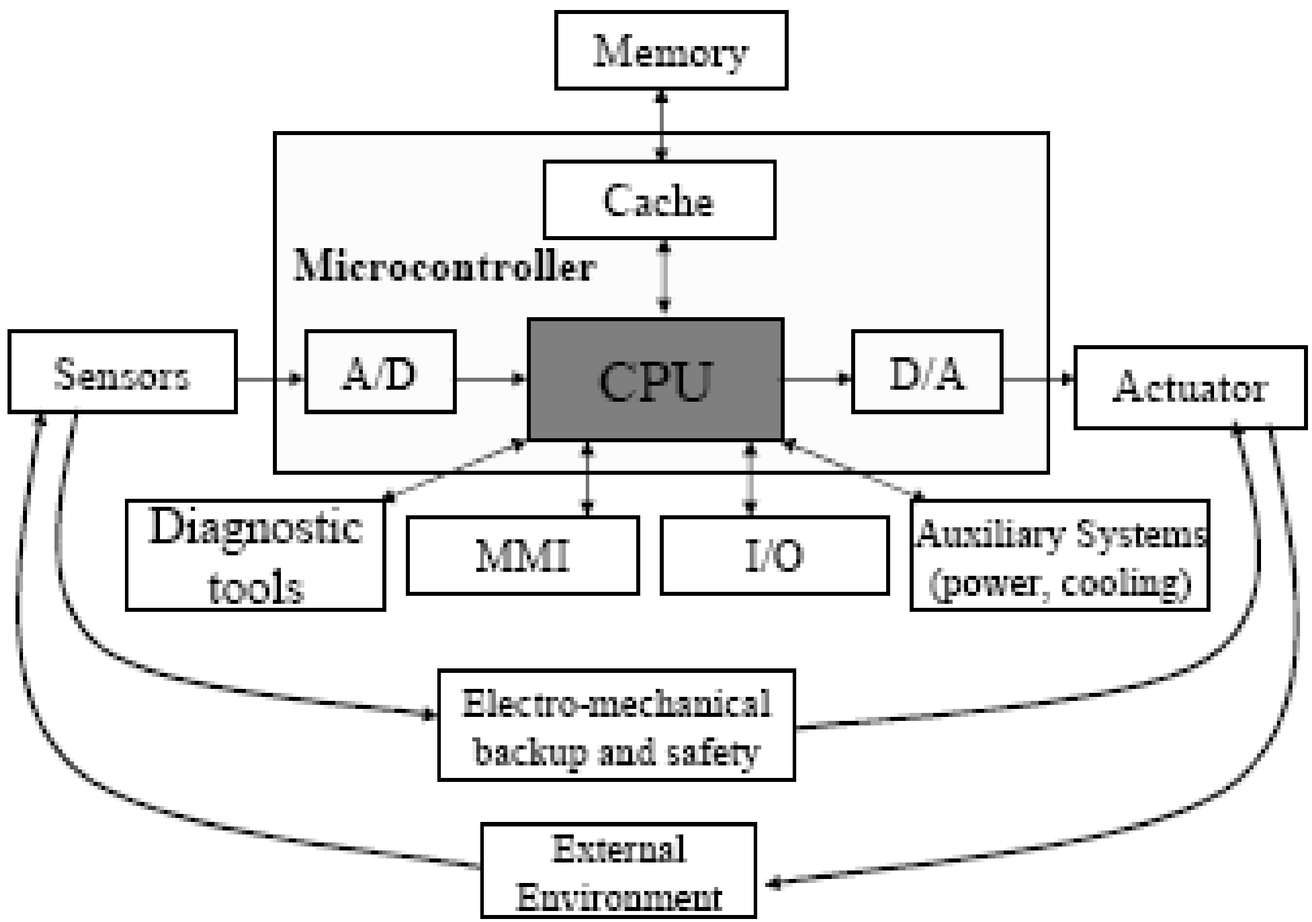
An Embedded Computer Designer's View

- Measured by: Cost, I/O connections, Memory Size, Performance

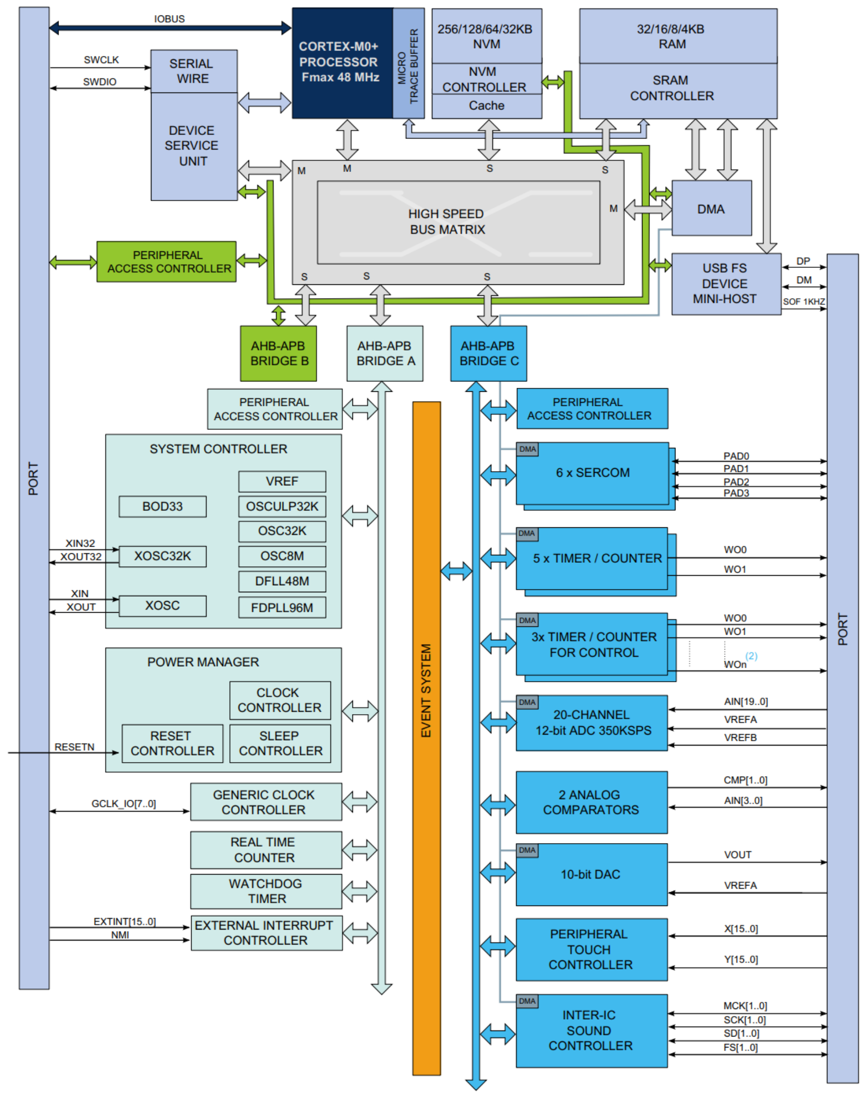


An Embedded Control System Designer's View

- Measured by: Cost, Time to Market, Cost, Functionality, Cost & Cost

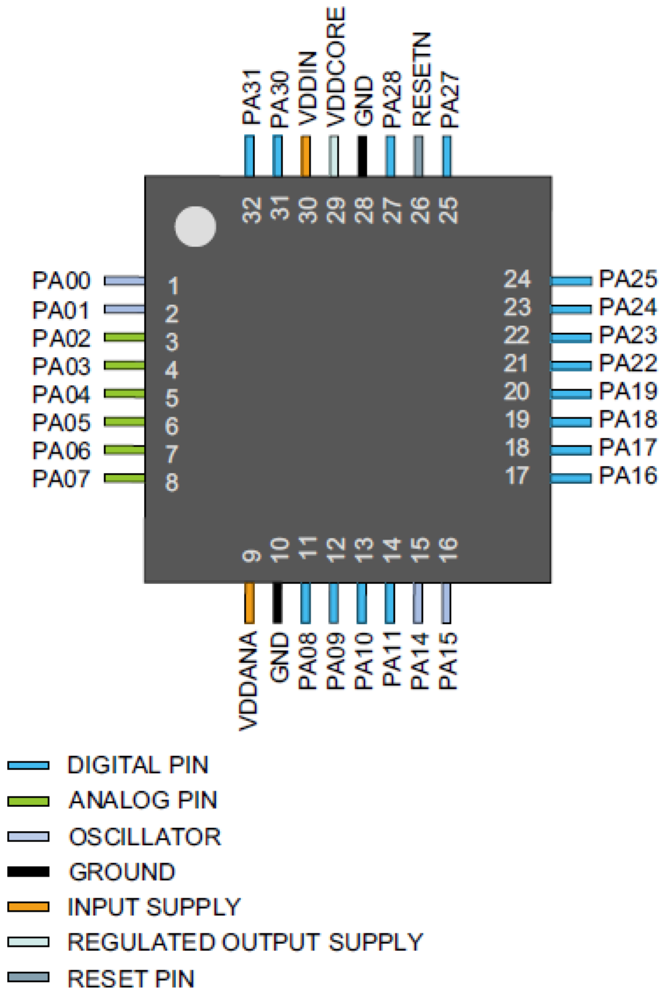


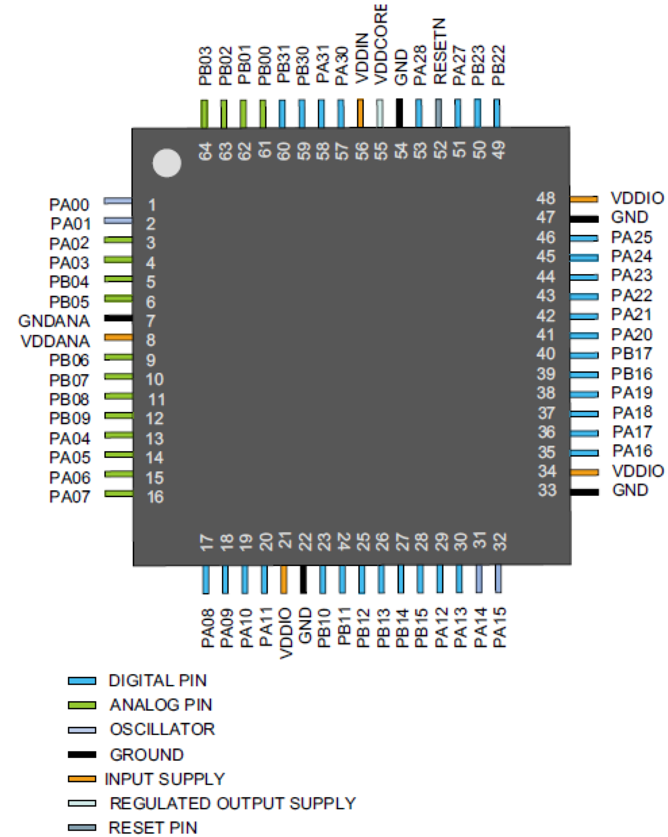
Atmel | SMART SAM D21 Processor (ARM Cortex M0+)



Atmel | SMART SAM D21 Processor (ARM Cortex M0+)

- Data sheet (All 1100 Pages!)





A Customer's View

- Reduced Cost
- Increased Functionality
- Improved Performance
- Increased Overall Dependability

Why are Embedded Systems Different?

Four General Categories of Embedded Systems

1. General Computing
   - Applications similar to desktop computing, but in an embedded package
   - Video games, set-top boxes, wearable computers, automatic tellers
   - Tablets, Phablets

2. Control Systems
   - Closed loop feedback control of real-time system
   - Vehicle engines, chemical processes, nuclear power, flight control
3. Signal Processing
   - Computations involving large data streams
   - Radar, Sonar, Video compression
4. Communication & Networking
   - Switching and information transmission
   - Telephone system, Internet
   - Wireless everything

Typical Embedded System Constraints

- Small Size, Low Weight
  - Handheld electronics
  - Transportation applications weight costs money

- Low Power
  - Battery power for 8+ hours (laptops often last only 2 hours)
  - Limited cooling may limit power even if AC power available

- Harsh environment
  - Heat, vibration, shock
  - Power fluctuations, RF interference, lightning
  - Water, corrosion, physical abuse

- Safety critical operation
  - Must function correctly
  - Must not function incorrectly
- Extreme cost sensitivity
  - $0.05 adds up over 1,000,000 units

Embedded System Design World-View

A complex set of tradeoffs:

- Optimize for **more than just speed**
- Consider **more than just the computer**
- Take into account **more than just initial product design**

Multi-Discipline

- Electronic Hardware
- Software
- Mechanical Hardware
- Control Algorithms
- Humans
- Society/Institutions

X

Multi-Phase

- Requirements
- Design
- Manufacturing
- Deployment
- Logistics
- Retirement

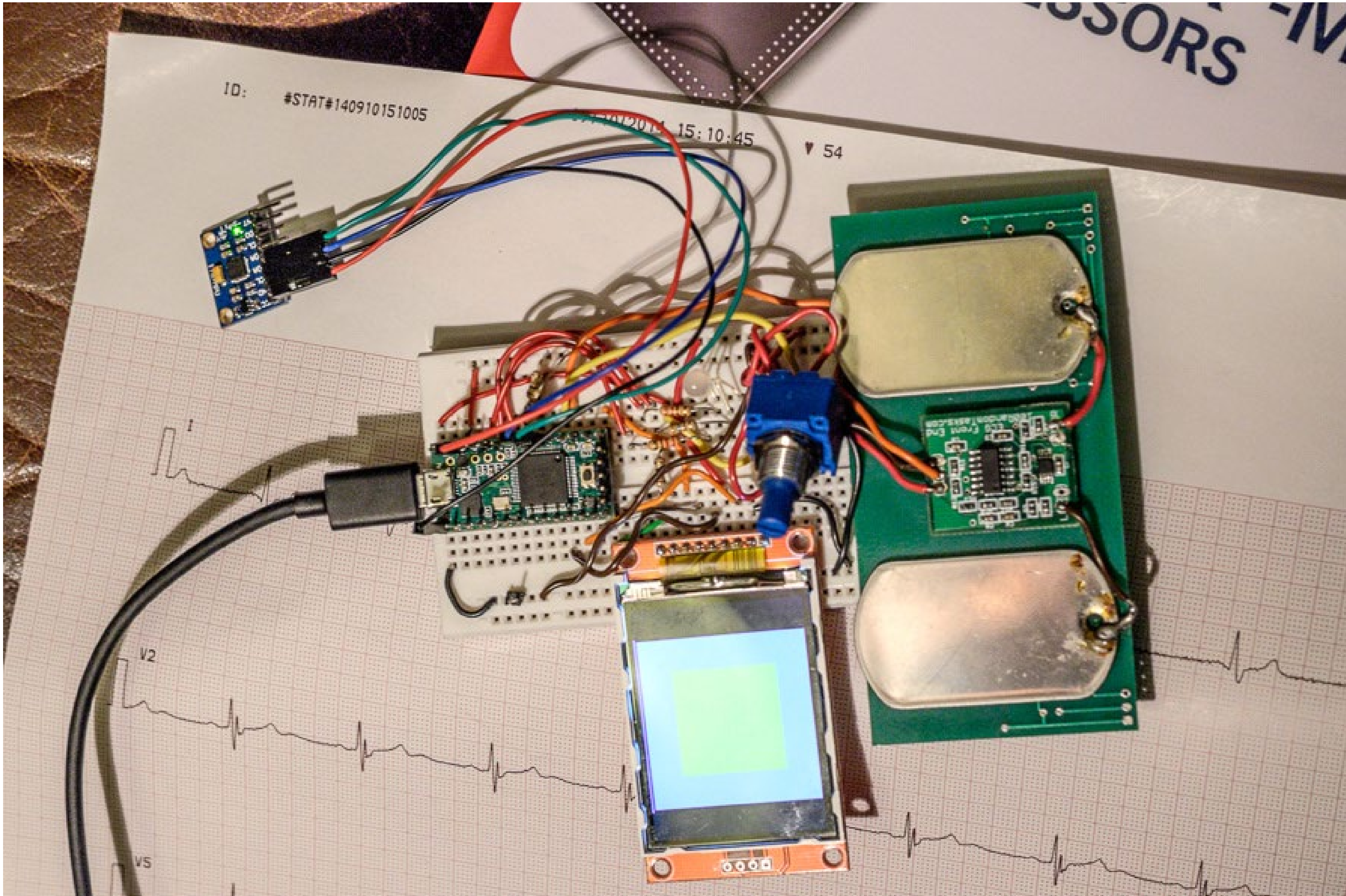
X

Multi-Objective

- Dependability
- Affordability
- Safety
- Security
- Scalability
- Timeliness

Embedded System Designer Skill Set

- Appreciation for multidisciplinary nature of design
  - Both hardware & software skills
  - Understanding of engineering beyond digital logic
  - Ability to take project from specification through production
- Communication and Teamwork skills
  - Work with other disciplines, manufacturing, marketing
  - Work with customers to understand the real problem being solved
  - Make a good presentation; even better – write "trade rag" articles
- And, by the way, technical skills too…..
  - Low-level: Microcontrollers, FPGA/ASIC, assembly language, A/D, D/A
  - High-Level: Object oriented design, C/C++, Real Time Operating Systems
  - Meta-level: Creative solutions to highly constrained problems
  - Likely in the future: **U**nified **M**odeling **L**anguage, embedded networks


ID:
#STAT#140910151005
15:10:45
54
V2
V5

Microprocessor or Microcontroller?

A Little History

What is a computer?

- One that computes; specifically: programmable electronic device that can store, retrieve and process data (from Merriam-Webster Dictionary)
- A computer is a machine that manipulates data according to a list of instructions (from Wikipedia)

Classifications of Computers:

- Personal computers
- Mainframes
- Supercomputers
- Dedicated controllers – Embedded controllers

MAINFRAMES

- Massive amounts of memory
- Use large data words – 64 bits or greater
- Mostly used for military defense and large business data processing
- Examples: IBM 4381; Honeywell DPS8

Personal Computers

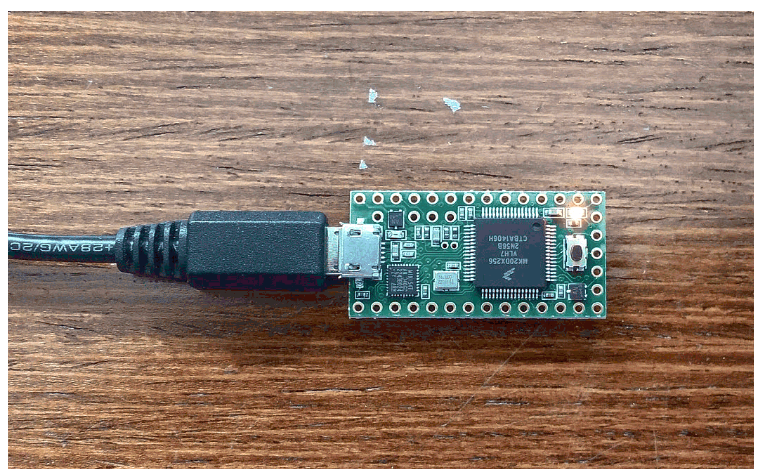
- Any general-purpose computer
  - Intended to be operated directly by an end user
- Range from small microcomputers that work with 4-bit words to PCs working with 32-bit words or more
- They contain a ***Processor*** – called different names
  - Microprocessor – built using Very-Large-Scale Integration technology; the entire circuit is on a single chip
  - Central Processing Unit (CPU)
  - Microprocessor Unit (MPU) – similar to CPU

Supercomputers

- Fastest and most powerful mainframes
  - Contain multiple central processors (CPU)
  - Used for scientific applications and number crunching
    - Now have petaflops performance – **FL**oating-point **O**perations **P**er **S**econd (FLOPS)
    - Used to measure the speed of the computer
- Examples of special-purpose supercomputers:
  - Belle, Deep Blue, and Hydra, for playing chess
  - GRAPE, for astrophysics and molecular dynamics
  - Deep Crack, for breaking the DES cipher
  - MDGRAPE3, for protein structure computation

Microcontrollers – Embedded Systems

- An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions often with real-time
- An integrated device which consists of multiple devices
  - Microprocessor(MPU)
  - Memory
  - I/O (Input/Output) ports
- Often has its own dedicated software



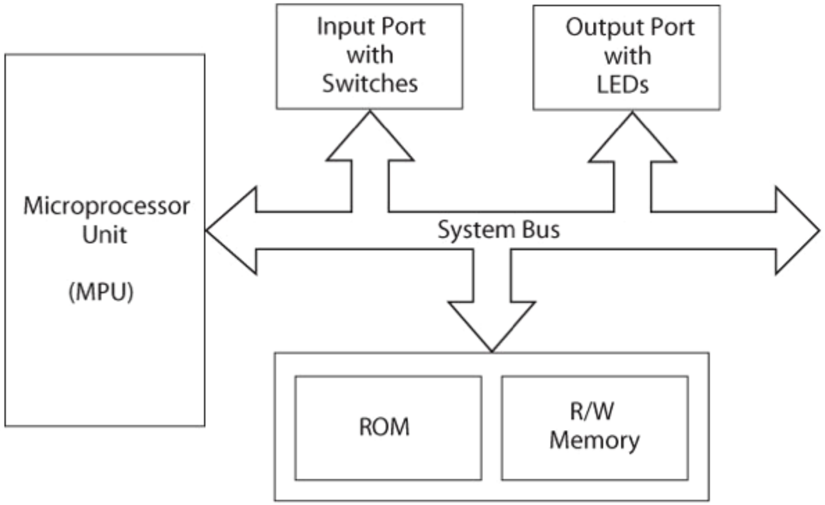
A little about
Microprocessor-based Systems ….

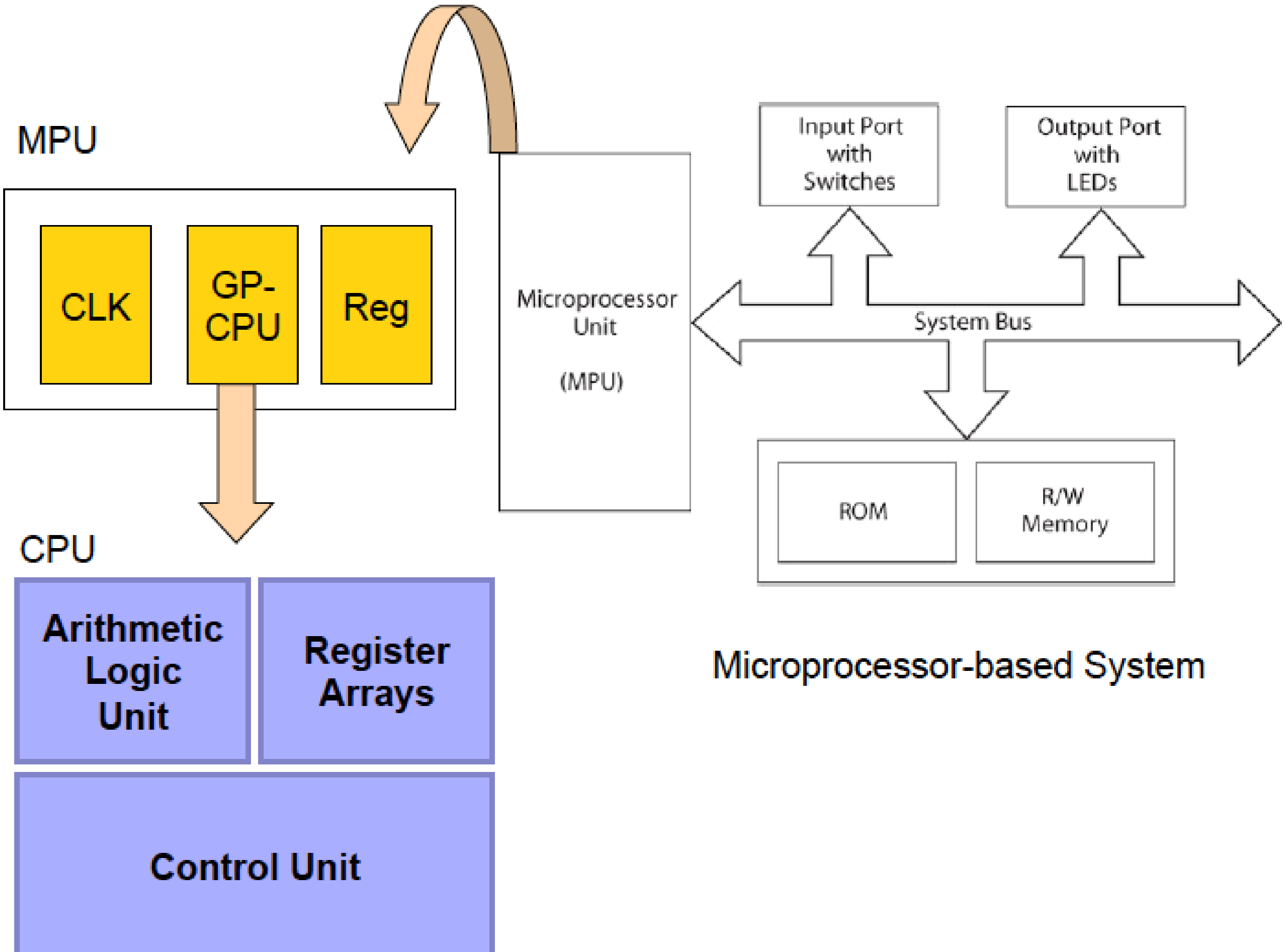
Evolution

- First came transistors
- Integrated circuits
  - SSI (Small-Scale Integration) to ULSI
  - Very Large Scale Integration circuits (VLSI)
- 1 – Microprocessors (MPU)
  - Microcomputers (with CPU being a microprocessor)
  - Components: Memory, CPU, Peripherals (I/O)
- 2 – Microcontroller (MCU)
  - Microcomputers (with CPU being a microprocessor)
  - Many special function peripheral are integrated on a single circuit
  - Types: General Purpose or Embedded System (with special functionalities)

Microprocessor-Based Systems

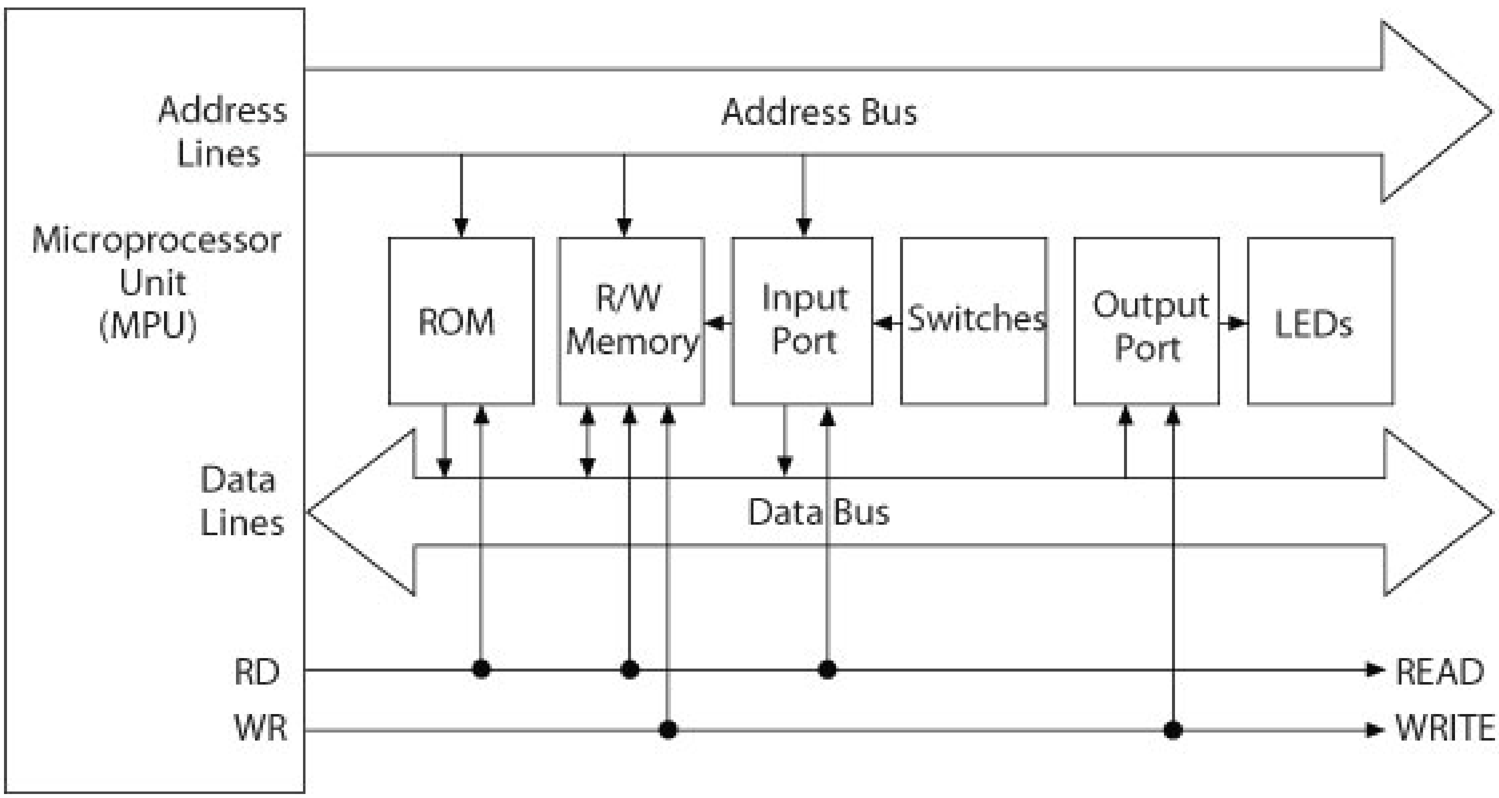
- Central Processing Unit (CPU)
- Memory
- Input/Output (I/O) circuitry
- Buses
  - Address bus
  - Data bus
  - Control bus





Microprocessor-based System

Microprocessor-Based System with Buses: Address, Data and Control



Microprocessor-Based Systems

Microprocessor

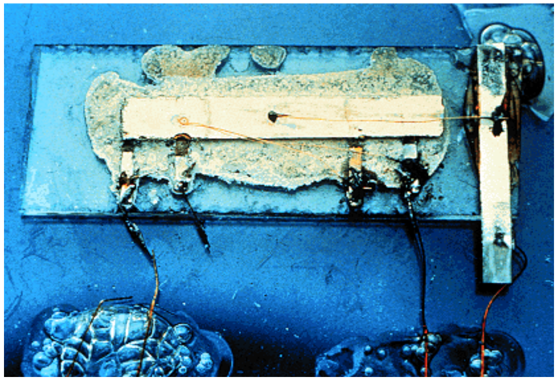
- The microprocessor (MPU) is a computing and logic device that executes binary instructions in a sequence stored in memory
- Characteristics:
  - General purpose central processing unit (CPU)
  - Binary
  - Register-based
  - Clock-driven
  - Programmable

The Evolution of CPUs

Transistors

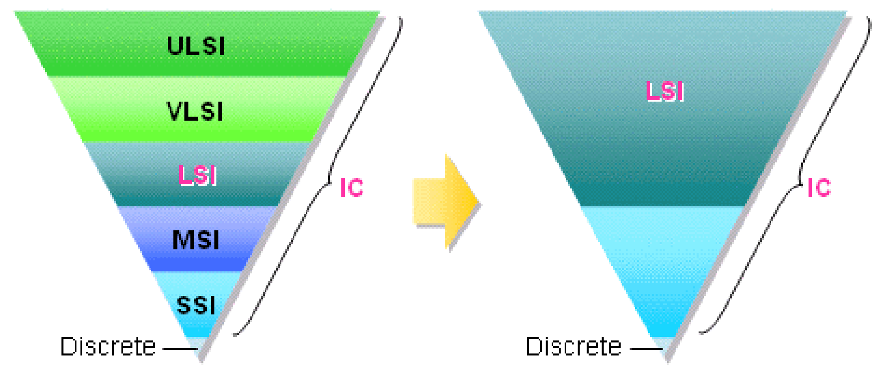
- Vacuum Tube: A device to control, modify, and amplify electronic signals
- Then came Transistors
  - Designed by John Bardeen, Will Shockley, and walter Brattain – scientists at the Bell Telephone Laboratories in Murray Hill, New Jersey, 1947
- In 1960, Jack Kilby and Robert Noyce designed the first integrated circuit (IC)
- Fairchild company manufactured logic gates

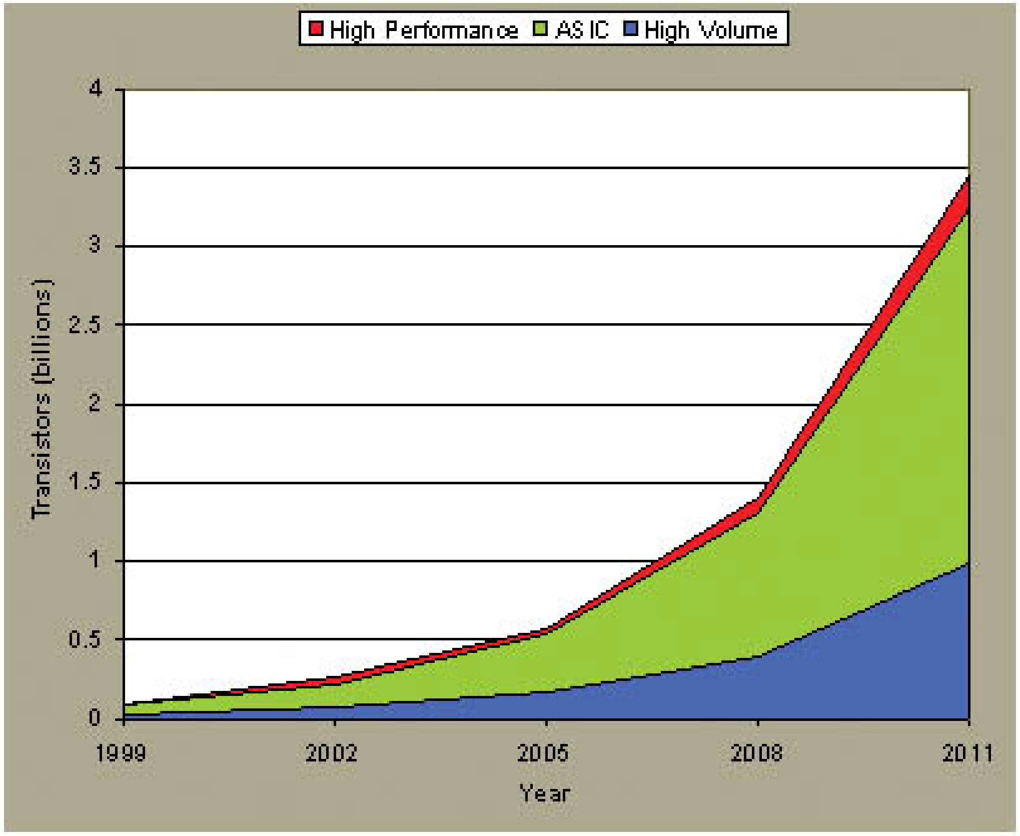
From Computer Desktop Encyclopedia
Reproduced with permission.
© 2000 Texas Instruments, Inc.



Integrated Circuits

- Advances in manufacturing allowed packing more transistors on a single chip
- Transistors and Integrated Circuits from SSI(Small-Scale Integration) to ULSI
- Birth of a microprocessor and its revolutionary impact





Microprocessors

- Noyce and Gordon Moore started Intel
- Intel designed the first calculator
- Intel designed the first programmable calculator
- Intel designed the first microprocessor in 1971
  - Model 4004
  - 4-bit; 2300 transistors, 640 bytes of memory, 108 KHz clock speed

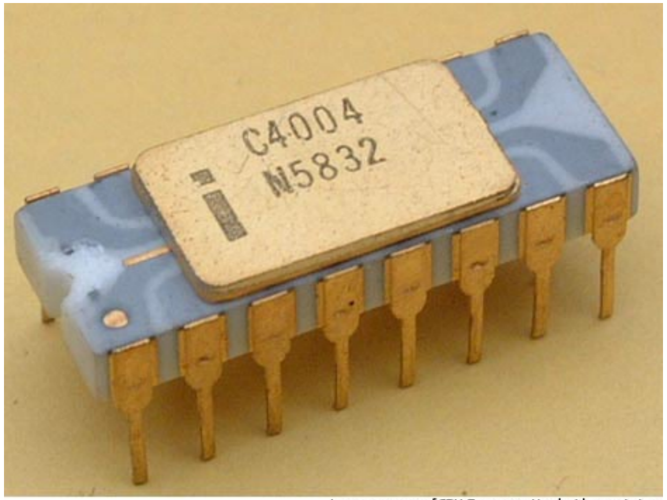
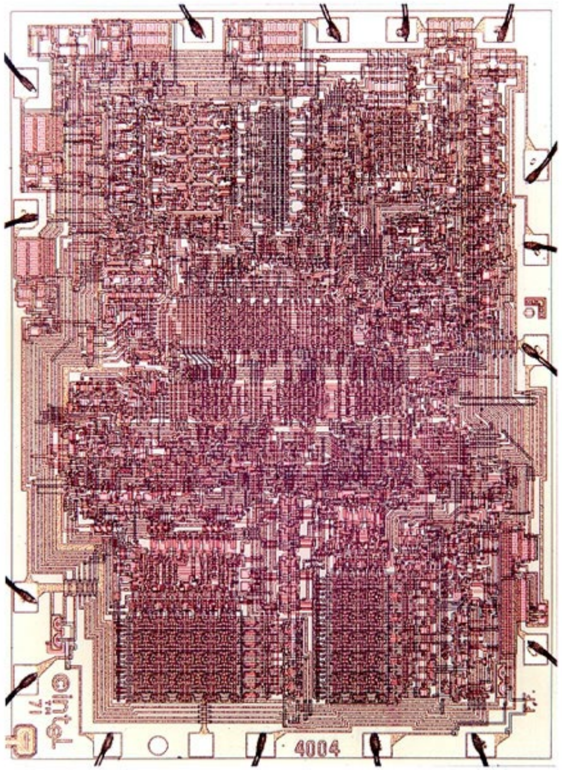


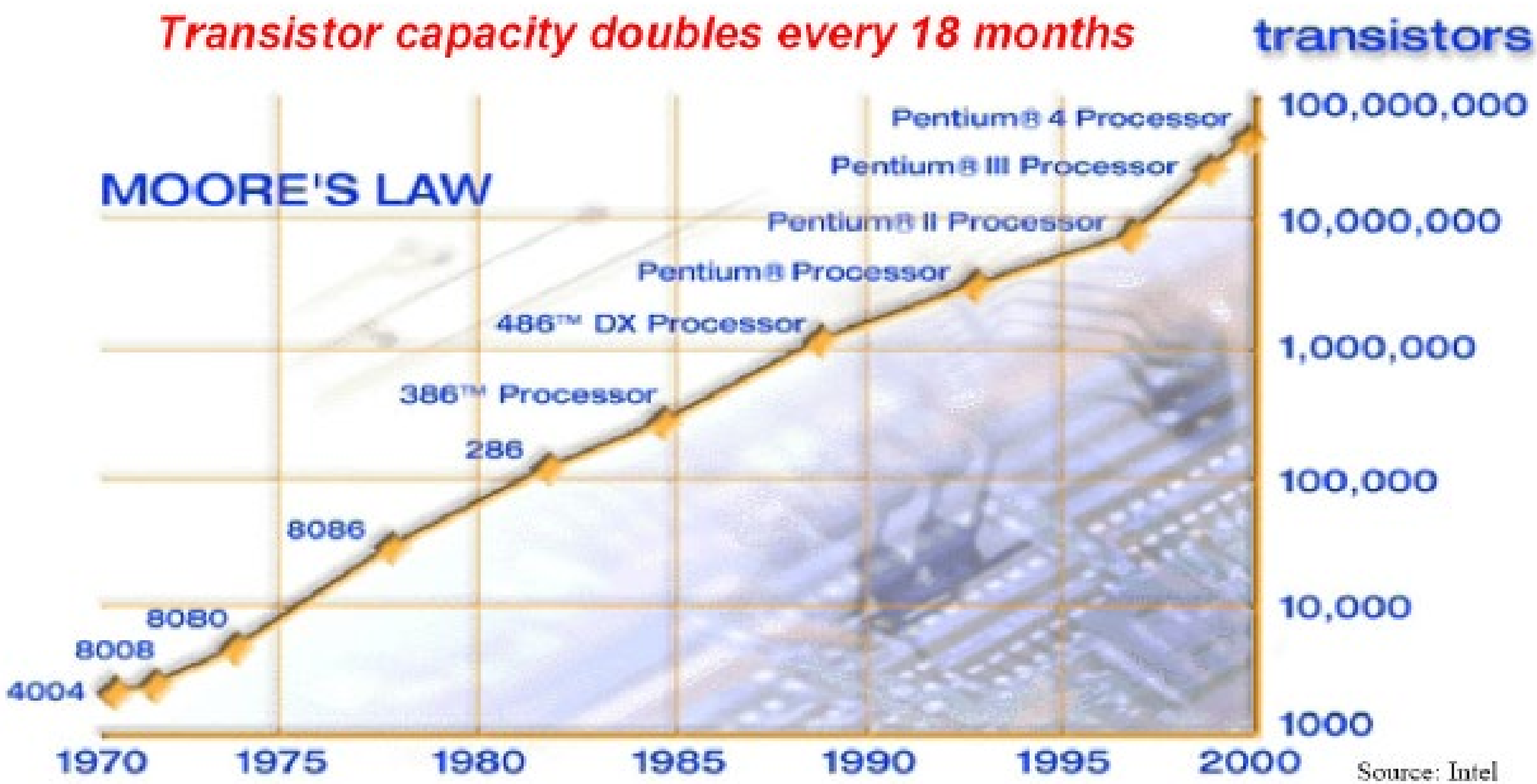
Image courtesy of CPU-Zone.com. Used with permission.



First Processors

- Intel released the 8086, a 16-bit microprocessor, in 1978
- Motorola followed with the MC68000 as their 16-bit processor
  - The 16-bit processor works with 16 bit words rather than 8 bit words
  - Instructions executed faster
  - Provide single instructions for more complex instructions such as multiply and divide
- 16-bit processors evolved into 32-bit processors
- Intel released the 80386
- Motorola released the MC 68020

Evolution of CPUs



In 1965, Gordon Moore, co-founder of Intel, indicated that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for the foreseeable future.

Evolution of CPUs

- Intel®Core™i7
  - Intel®Core™i7-5960X Processor Extreme Edition
  - 20M Cache, up to 3.50 GHz
  - 8 Cores, 16 Threads
  - 64-bit Instruction Set

Microprocessor-based Systems

Memory Types

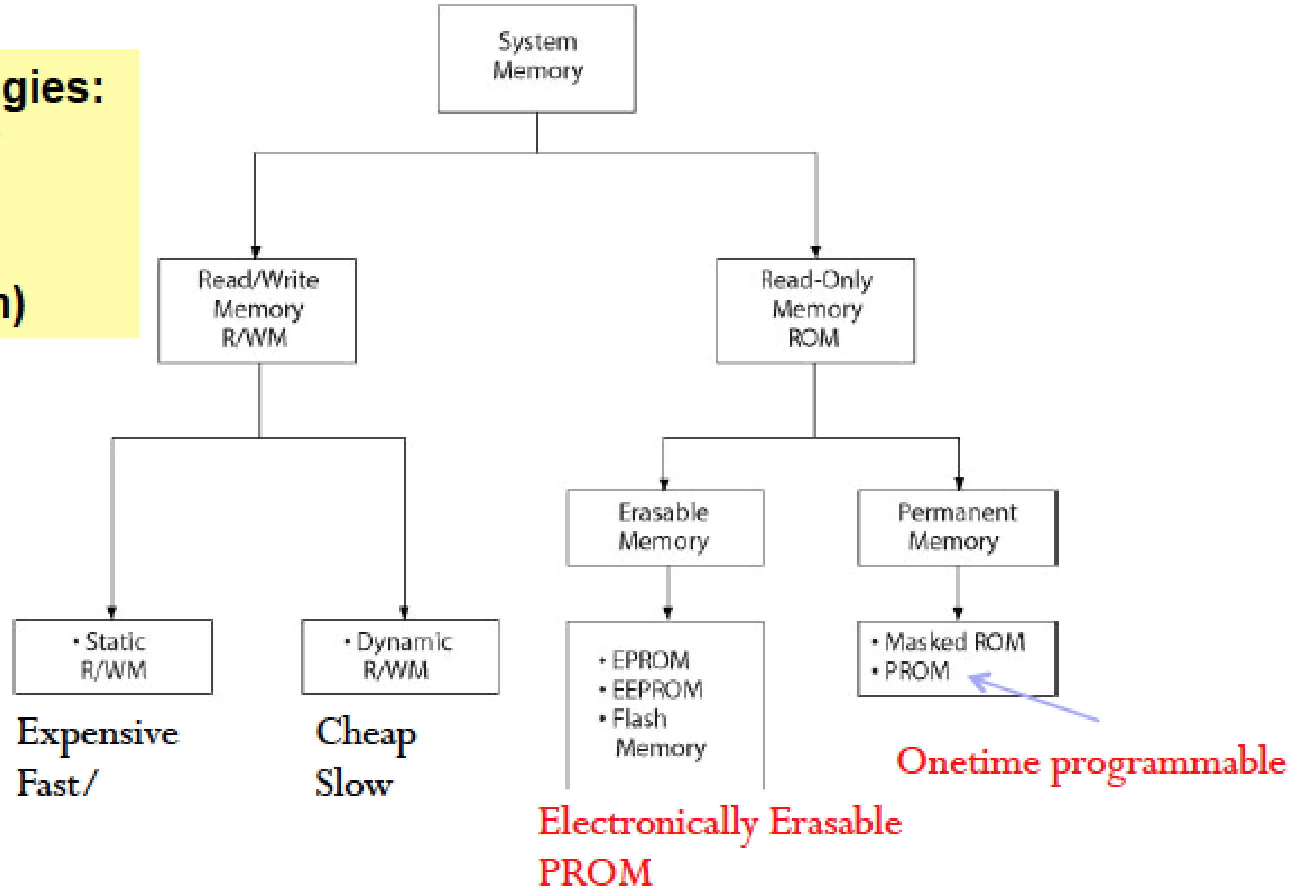
- **R/W: Read/Write Memory; also called RAM**
  - It is volatile (losses information as a power is removed)
  - Write means the processor can store information
  - Read means the processor can retrieve information from the memory
  - Acts like a Blackboard!

- **ROM: Read-Only Memory**
  - It is typically non-volatile (permanent) – can be erasable
  - It is similar to a page from your textbook

Microprocessor-Based Systems

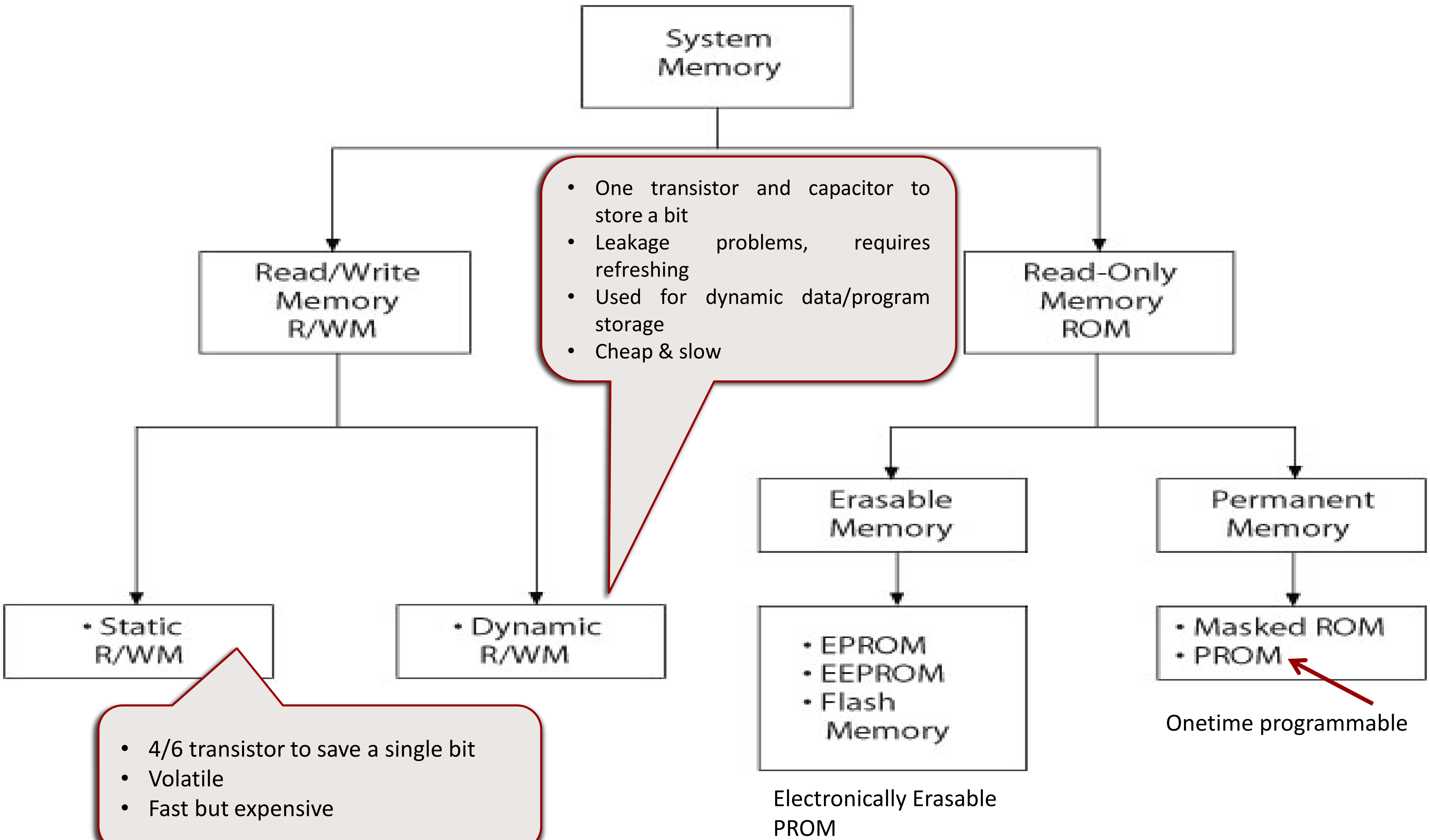
Memory Classification

Basic Technologies:
Semiconductor
Magnetic
Optical
(or combination)



Microprocessor-Based Systems

Memory Classification



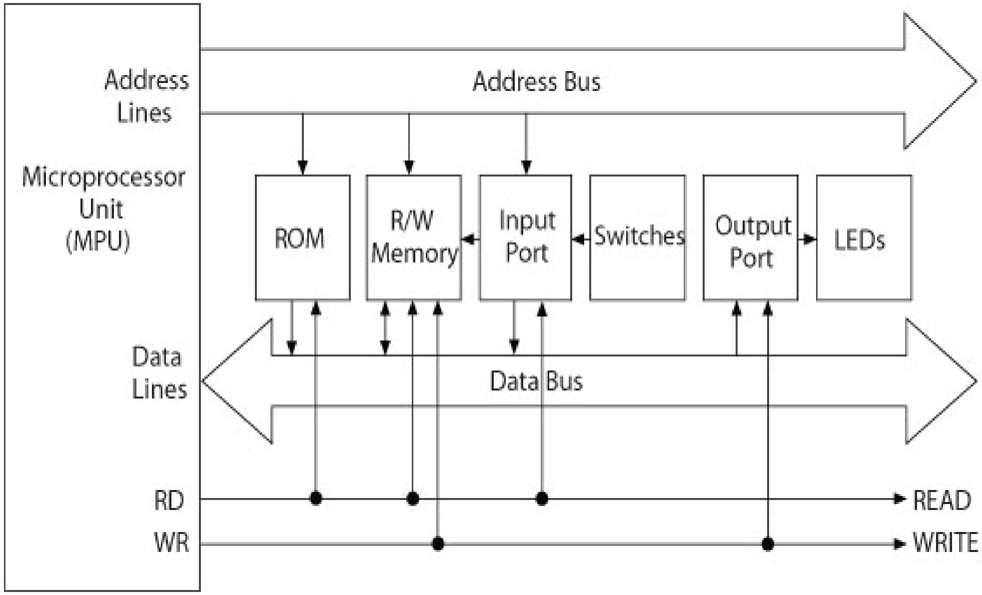
Erasable ROMs

- Masked Programmable ROM
  - Programmed by the manufacturer
- Programmable ROM (PROM)
  - Can be programmed in the field via the programmer
- Erasable Programmable ROM (EPOM)
  - Uses ultraviolet light to erase (through a quartz window)
  - OTP refers to One-Time Programmable
- Electrically Erasable Programmable ROM (EEPROM)
  - Each program location can be individually erased
  - Expensive
  - Requires programmer
- FLASH
  - Can be programmed in-circuit (in-system)
  - Easy to erase (no programmer)
  - Only one section can be erased/written at a time (typically 64 bytes at a time)

Microprocessor-based Systems

I/O Ports

- The way the computer communicates with the outside world devices
- I/O ports are connected to Peripherals
  - Peripherals are I/O devices
    - Input devices
    - Output devices
  - Examples
    - Printers and modems
    - Keyboard and mouse
    - Scanner
    - Universal Serial Bus (USB)



Microprocessor-based Systems

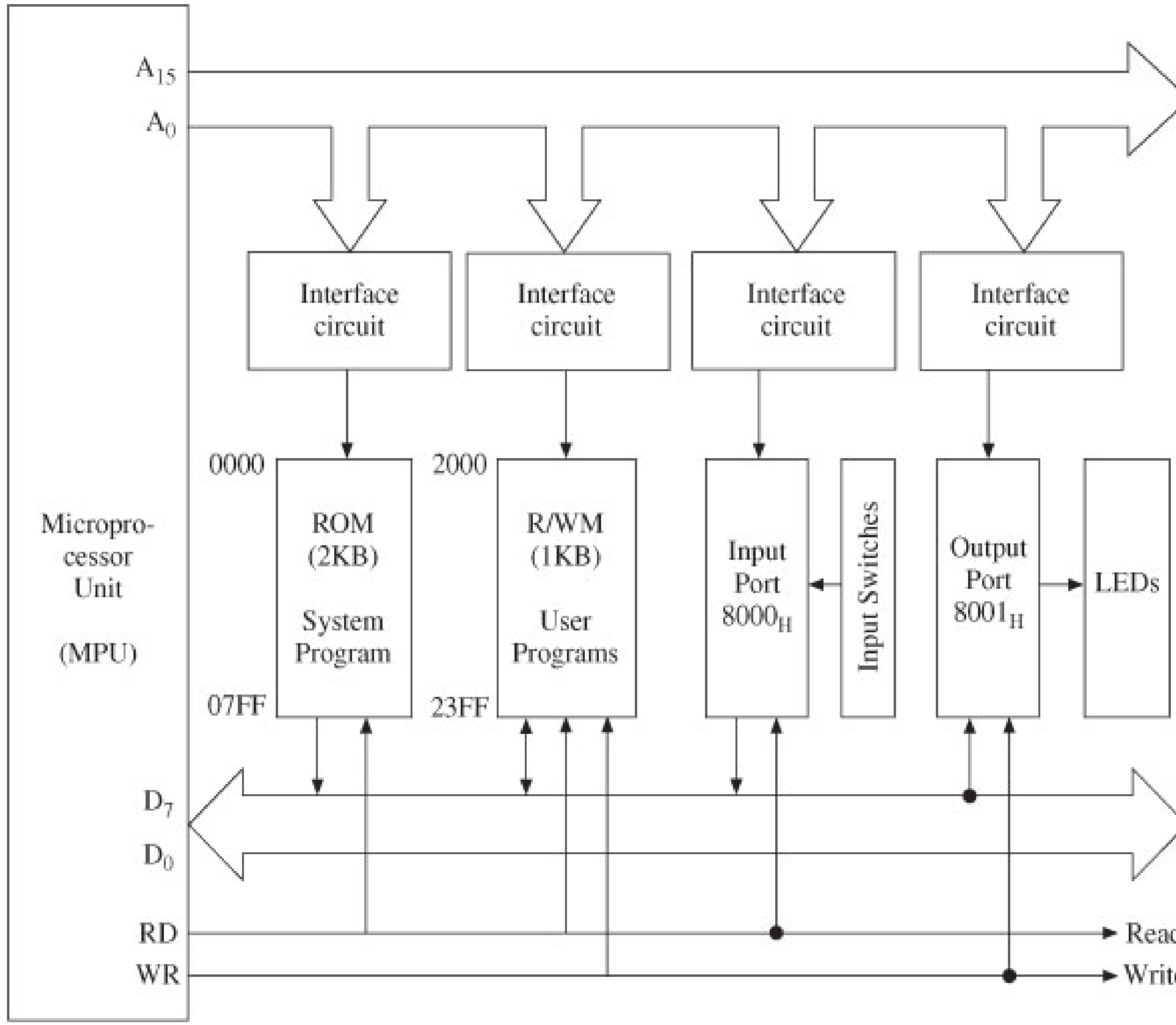
BUS

- **The three components – MPU, memory, and I/O, are connected by a BUS**

- **Address Bus**
  - Consists of 16, 20, 24, or 32 parallel lines (wires) – unidirectional
  - These lines contain the address of the memory location to read or written
- **Control Bus**
  - Consists of 4 to 10 (or more) parallel signal lines
  - CPU sends signals along these lines to memory and to I/O ports
    - Examples: Memory Read, Memory Write, I/O Read, I/O Write
- **Data Bus**
  - Consists of 8, 16, or 32 parallel lines
  - Bi-directional
  - Only one device at a time can have its outputs enabled
  - This requires the devices to have three-state output

Expanded Microprocessor-Based System

- Note the direction of the busses.
- What is the width of the address bus?
- What is the value of the Address bus to access the first register of the R/WM?



Remember: 111 1111 1111 = 2^11=2K

Now what about Microcontrollers???

First Microcontrollers

- IBM started using Intel processors in its PC
  - Intel started its 8042 and 8048 (8-bit microcontroller) – using in printers
- Apple Macintosh used Motorola 68000
- In 1980 Intel abandoned microcontroller business
- By 1989, Microchip was a major player in designing microcontrollers
  - PIC: Peripheral Interface Controller

Embedded Controllers

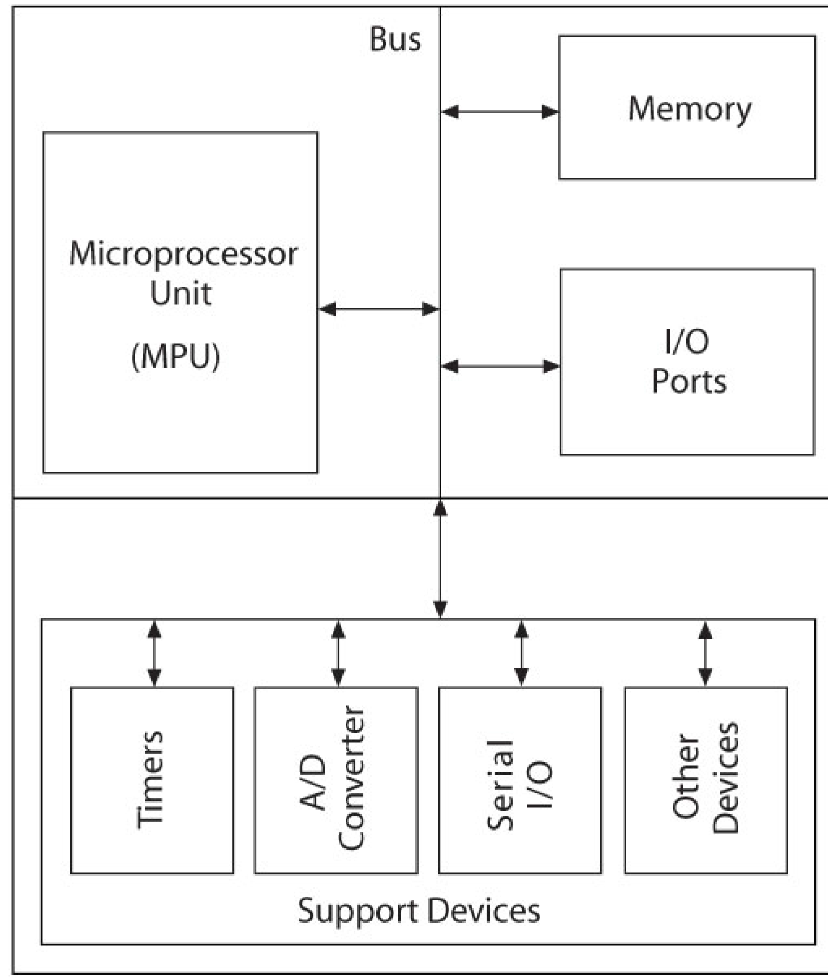
Software Characteristics

- No operating systems
- Execute a single program, tailored exactly to the controller hardware
- Assembly language (vs. High-level language)
  - Not transportable, machine specific
  - Programmer needs to know CPU architecture
  - Speed
  - Program size
  - Uniqueness

Microcontroller Unit (MCU)

Block Diagram

- An integrated electronic computing and logic device that includes three major components on a single chip
  - Microprocessor
  - Memory
  - I/O ports
- Includes support devices
  - Timers
  - A/D converter
  - Serial I/O
  - Parallel Slave Port
- All components connected by common communication lines called the system bus



MCU Architecture

- RISC (Harvard)
  - Reduced instruction set computer
  - Simple operations
  - Simple addressing modes
  - Longer compiled program but faster to execute
  - Uses pipelining
- CISC (Von Neuman)
  - Complex instruction set computer
  - More complex instructions (closer to high-level language support)

Bench marks: How to compare MCUs together
MIPS: Million Instructions / second (Useful when the compilers are the same)

Main 8-bit Controllers

- Microchip -PIC® Microcontrollers
  - RISC architecture (reduced instruction set computer)
  - Has sold over 2 billion as of 2002
  - Cost effective and rich in peripherals
- Motorola – now Freescale
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 68HC05, 68HC08, 68HC11
- Intel – now Marvell
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 8051, 8052
  - Many different manufacturerers: Philips, Dallas/MAXIM Semiconductor, etc.
- Atmel
  - RISC architecture (reduced instruction set computer)
  - Cost effective and rich in peripherals
  - AVR

Software: From Machine to High-Level Languages

(1 of 3)

High-Level Language
Assembly Language
Machine Language

- Machine Language: binary instructions
  - All programs are converted into the machine language of a processor for execution
  - Difficult to decipher and write
  - Prone to cause many errors in writing

Software: From Machine to High-Level Languages

(2 of 3)

- Assembly Language: machine instructions represented in mnemonics
    - Has one-to-one correspondence with machine instructions
    - Efficient in execution and use of memory; machine-specific and not easy to troubleshoot

High-Level Language
Assembly Language
Machine Language

Software: From Machine to High-Level Languages

(3 of 3)

- High-Level Languages: Such as BASIC, C, and C++
  - Written in statements of spoken languages (such as English)
    - Machine independent
    - Easy to write and troubleshoot
    - Requires large memory and less efficient in execution

High-Level Language
Assembly Language
Machine Language

Data Format (8-bit)

(1 of 4)

- Unsigned Byte: All eight bits (Bit0 to Bit7) represent the magnitude of a number
  - Range 0 to FF in Hex and 0 to 255 in decimal

Unsigned

Signed

Data Format (8-bit)

(2 of 4)

- Signed Byte: Seven bits (Bit0 to Bit6) represent the magnitude of a number
  - The eighth bit (Bit7) represents the sign of the number. The number is positive when Bit7 is zero and negative when Bit7 is one.
  - Positive Numbers: 0 to 7F (0 to 127)
  - Negative Numbers: 80 to FF (-1 to -128)
  - All negative numbers are represented in **2's compliment**

Unsigned

Signed

Data Format (8-bit)

(3 of 4)

- Binary Coded Decimal Numbers (BCD)

  - 8 bits of a number divided into groups of four, and each group represents a decimal digit from 0 to 9
  - Four-bit combinations from A through F in Hex are invalid in BCD
  - Example: **0010 0101** represents the binary coding of the decimal number 25d which is different in value from 25H

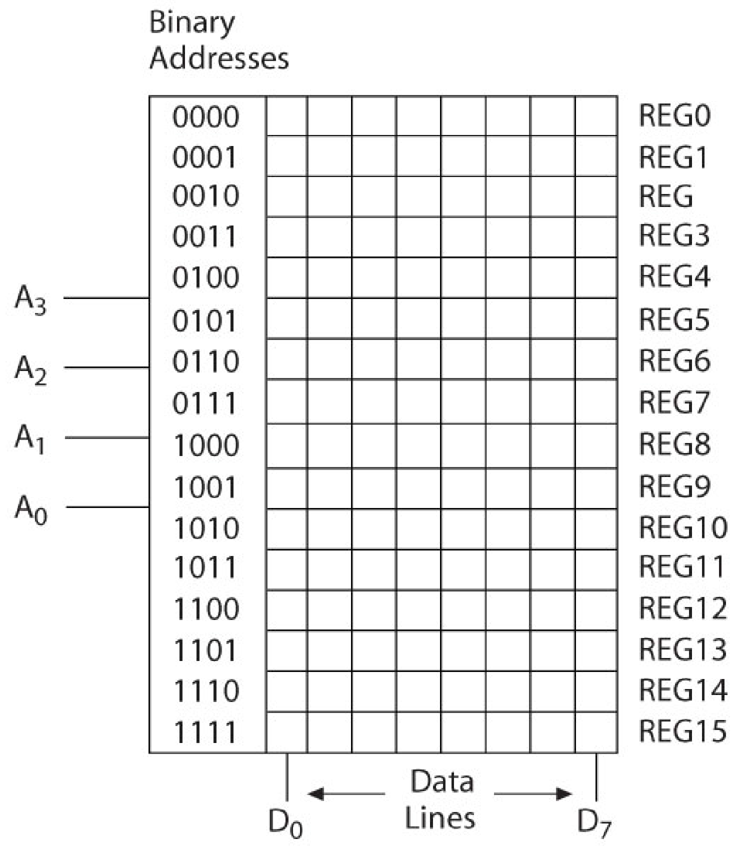
Data Format (8-bit)

(4 of 4)

- American Standard Code for Information Interchange (ASCII)
  - Seven-bit alphanumeric code with 128 combinations (00 to FF)
  - Represents English alphabet, decimal digits from 0 to 9, symbols, and commands

Storing Bits in Memory

- We can store in different memory types
  - EEPROM, FLASH, RAM, etc.
- In an 8-bit RAM
  - Each byte is stored in a single memory register
  - Each word is stored in two memory locations (registers)
  - DATA 0x1234
    - 0x12 ➔REG11 (High-order byte)
      - 0001 0010
    - 0x34➔REG10 (Low-order byte)
      - 0011 0100

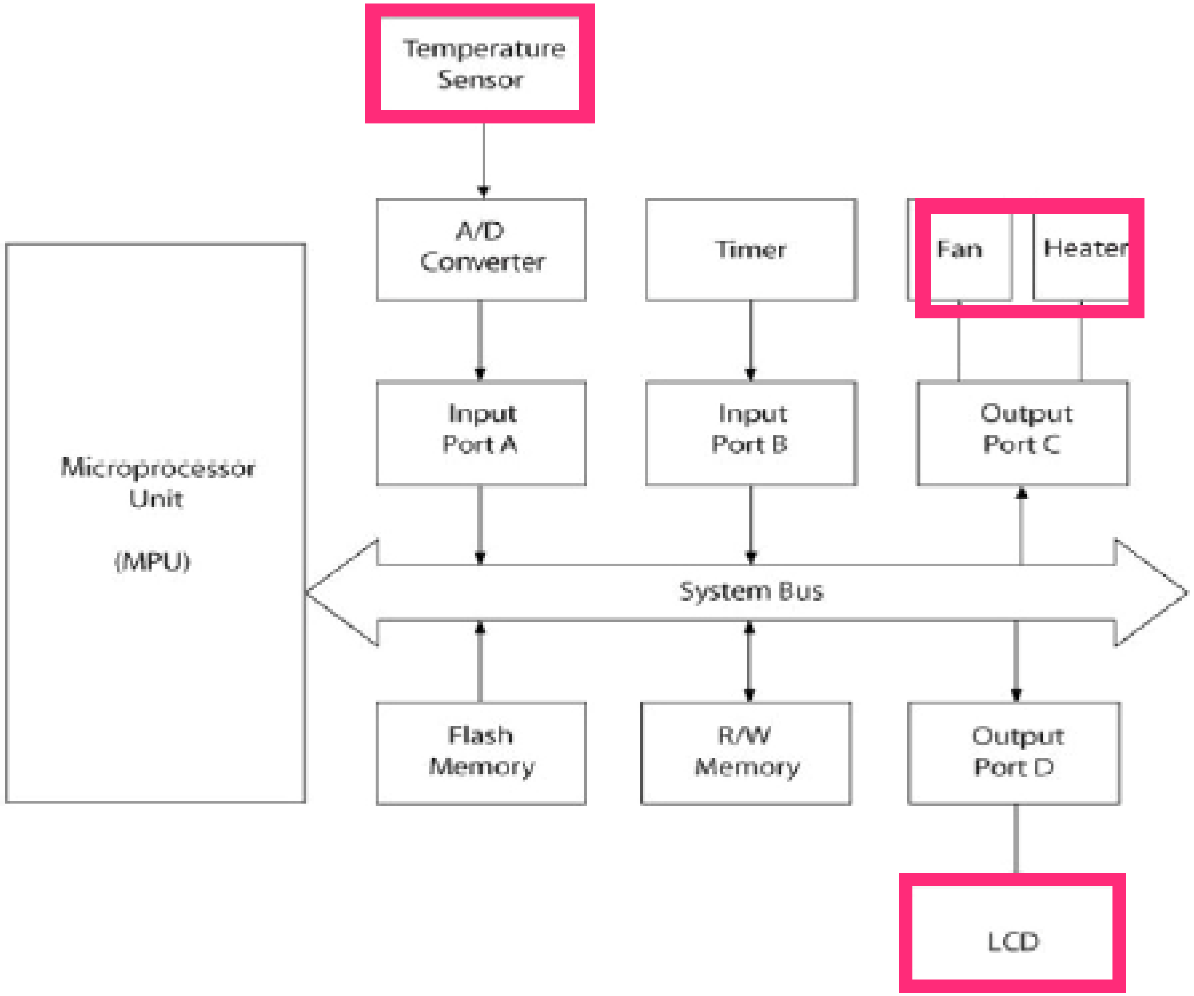


Remember: -8 -> 111 1000 (in two's complement)

Design Examples .....

Microcontrollers vs. Microprocessors

MPU-Based Time and Temperature System



MCU-Based Time and Temperature System